Zero-Shot Learning Through Cross-Modal Transfer

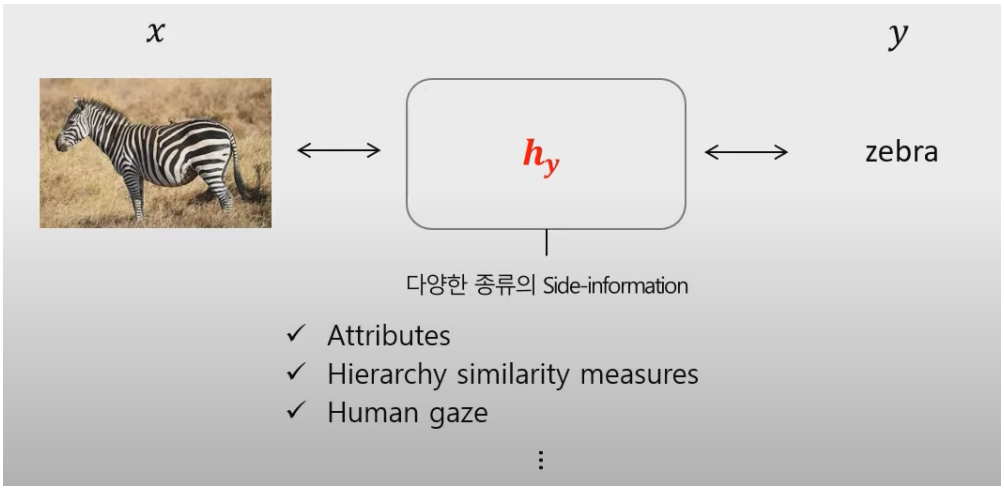
> Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C. D., & Ng, A. Y. (2013). *Zero-Shot Learning Through Cross-Modal Transfer*
> **Reviewer: 고경현**

Zero-shot learning의 개요와 해당 논문에서 제안한 method를 리뷰합니다. 논문의 흐름을 따르지만 논문에 등장하지 않은 내용들도 다소 포함될 수 있습니다.

1. Introduction

Zero-shot learning (ZSL)은 이미지 분류 시 training 단계에서 사용되지 않은 class를 예측하게 하는 학습 방법이다. 정보에 기반한 추론이 가능한 인간의 사고에 기초하여 발전했다. 예를 들어, '초록색, 빨간색, 동그란 과일'이라는 특징을 들었을 때 '사과' 혹은 '사과와 비슷한 것'이라고 추론이 가능한 것처럼 말이다.

기초적인 ZSL의 과정은 굉장히 직관적이다.



출처 : **[DMQA Open Seminar] Zero-shot learning @Youtube**

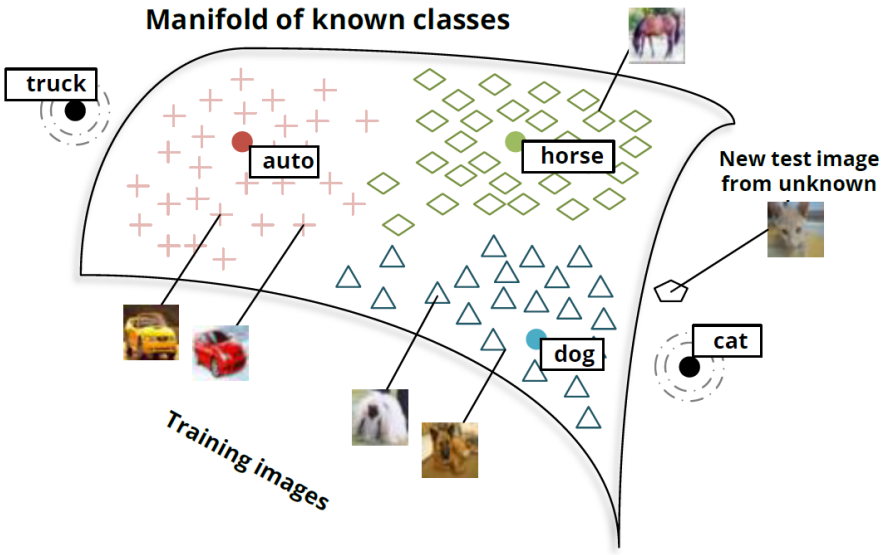
이미지의 피처를 그대로 활용하는 것이 아닌 피처로부터 Side information을 추출하여 그 공간에서 이미지를 분류한다. 이 중간 과정 덕분에 training에 사용되지 않은 이미지까지 분류할 수 있다. 본 논문에서는 워드 임베딩을 활용한 side information을 사용했다고 볼 수 있다.

training 단계에서 사용된 class를 *seen class,* 사용되지 않은 class를 *unseen class*라고 했을 때, 이 논문에서 제안하는 모델은 seen / unseen class 모두 예측이 가능한 모델이다. 다만 unseen class의 true class는 모델이 전혀 알 수 없는 정보이기 때문에, 가장 근접한 seen class로 예측한다.

논문에서 제안한 모델은 크게 두 가지 메인 아이디어를 기반으로 구성된다.

1. 이미지 피처를 해당 클래스 단어들의 semantic space로 매핑한다.
    - 예) 매핑 : '개' 이미지 피처 벡터 → '개' 단어 임베딩 벡터
2. test 이미지가 seen class인지 unseen class인지 이진 분류하는 novelty detection이 이루어진다.
    - 학습된 분류기는 seen class로 분류하려는 경향이 있기 때문이다.
    - 본 논문에서 novelty detection은 후술할 두 가지 outlier detection 방법을 이용한다.

이를 도식화하면 아래와 같다. 검은색 테두리 내부인 manifold 내에 존재하는 이미지는 training 단계에서 학습된 seen class에 속하며, 밖에 있는 truck과 cat은 unseen class에 속한다. 이미지를 특정 공간에 매핑시킨 후 novelty detection을 통해 seen / unseen class를 분류하고 나서 class를 예측한다.



앞으로 모델이 이미지들을 manifold 상으로 어떻게 매핑시키는지와 매핑된 후 seen class와 unseen class를 어떻게 구분하는지에 집중하여 논문을 이해하면 좋을 것이다.

2. Related Work

관련된 선행 연구들을 살펴본 후에 모델을 알아보자.

1. Zero-Shot Learning

- Palatucci의 "Zero-Shot Learning with Semantic Output Codes"

  본 논문의 방법과 가장 유사한 방법이다. 사람들이 특정 단어를 생각하고 있을 때의 fMRI 이미지를 manually 하게 만든 semantic feature의 공간으로 매핑시키고, 이 매핑된 공간에서 단어 class를 예측하도록 했다. 이 방법을 통해 ZSL이 가능했다.

- Lampert의 "Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer"

입력된 이미지를 이미지의 시각적인 attributes로부터 manually하게 만들어진 binary variable feature space로 매핑하고, 이 매핑된 공간에서 class를 예측하는 방법을 사용했다.

위의 논문들처럼 해당 논문에서 등장하는 방법도 이미지를 특정 space로 매핑시켜 class를 예측한다. 하지만 위 논문들에서는 이미지를 다른 공간에 매핑시키기 위해 manually하게 만들어진 semantic/attribute space를 사용하는 반면, 본 논문에서는 비지도학습 방법에 의해 만들어진 semantic space를 사용한다는 것이 차이점이다.

2. One-Shot Learning

One-shot learning이란 training example이 매우 적은 경우에 사용하는 학습 방법이다.

- Salakhutdinov의 “Learning with Hierarchical-Deep Models”

  이 논문에서는 신경망 모델과 계층적 베이지안 모델을 결합하여 One-shot learning을 하는 이미지 분류 모델을 만들었다.

본 논문에서도 신경망 모형을 학습하고 이를 확률적 모델을 사용하여 knowledge transfer를 한다는 점이 비슷하다. 이에 더해 자연어로부터 cross-modal knowledge transfer를 하기 때문에 훈련 데이터가 필요 없다는 장점도 있다.

3. Knowledge and Visual Attribute Transfer

- Lampert, Farhadi 의 “Describing Objects by their Attributes”와 “Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer”

  위 두 논문에서는 unseen class를 분류하기 위해 아래 그림처럼 이미지의 시각적인 특징을 담는 manually하게 잘 만들어진 binary variable을 사용했다.

출처 : “Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer”

반면 본 논문에서는 이미지의 시각적인 특징이 아닌 비지도학습을 통해 학습된 단어의 분포적 특성(워드 임베딩)만 가지고 분류된다.

4. Domain Adaptation

Domain adaptation은 한 도메인에는 데이터가 많지만, 다른 도메인에는 그렇지 않을 때 유용한 방법이다. 예를 들어, 영화 리뷰 데이터로 학습된 감성 분석 분류기를 책 리뷰에 적용하는 것이 유용할 수 있으며, 이것이 바로 domain adaptation이다.

본 논문에서 제시한 방법을 통해 domain adaptation이 빠르게 이루어짐을 장점으로 꼽는다.

5. Multimodal Embeddings

Multimodal embedding은 여러 source에서 나온 데이터를 연결 짓는다. 예를 들어 비디오, 책, 사진 등에 '배(boat)'가 등장한다면 multimodal embedding을 통해 여러 곳에 등장한 배를 하나의 공통된 space에 매핑시킬 수 있다.

본 논문에서 등장한 모델은 시각적(visual) 이미지를 단어(text)로 매핑시켜 여러 source의 정보를 하나의 공간으로 매핑시키는 multimodal embedding의 특징과 비슷하다.

3. Methods

본 논문의 사용된 방법은 다음과 같다. 이미지의 feature vector를 해당 class의 semantic word vector로 매핑시킨다. 매핑된 벡터들은 semantic space 상에 존재하게 되며, 이 매핑된 공간 위에서 novelty detection을 통한 seen / unseen class 이진 분류가 먼저 일어난 후 seen class와 unseen class에 각각 다른 분류 모델을 적용하여 이미지 클래스를 분류한다.

1. Projecting Images into Semantic Word Spaces

디테일을 알아보자. 우선 이미지 피처 벡터를 semantic word space로 매핑하는 과정이 필요하다. 이미지 class가 갖는 시각적인 정보를 그 이미지의 class가 text에서 갖는 semantic한 정보로 바꾸는 cross-modal transfer가 일어난다는 것이다. 예를 들어, 고양이 이미지 피처 벡터를 '고양이'가 단어로서 갖고 있는 semantic 벡터로 매핑하는 것이다.

이 과정은 아래 식 $J(\Theta)$을 최소화하는 neural network parameter $\theta^{(1)}$, $\theta^{(2)}$을 찾는 것과 동일하다.

$$J(\Theta) = \sum_{y \in Y_s} \sum_{x^{(i)} \in X_y} ||w_y - \theta^{(2)} f(\theta^{(1)} x^{(i)})||^2$$

$$\min J(\Theta)$$

- $y \in Y_s$: seen class에 속하는 이미지 label
- $x^{(i)} \in X_y$: seen class y의 이미지 feature

  ▼ 추가 설명

  이 이미지 피처는 Coates의 "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization"에 등장한 비지도학습 이미지 피처 추출 방식을 통해 추출된 12800차원의 벡터다.
- w_y : 위키피디아 text 데이터를 이용해 비지도학습에 의해 만들어진 50차원 단어 벡터

  ▼ 추가 설명

  이 벡터를 만든 모델은 각각의 단어가 속하는 context로부터 해당 단어가 등장할 가능도를 예측함으로써 만들어졌다. 또한 이 모델은 단어의 local context와 global context를 모두 사용하여 벡터를 만들기 때문

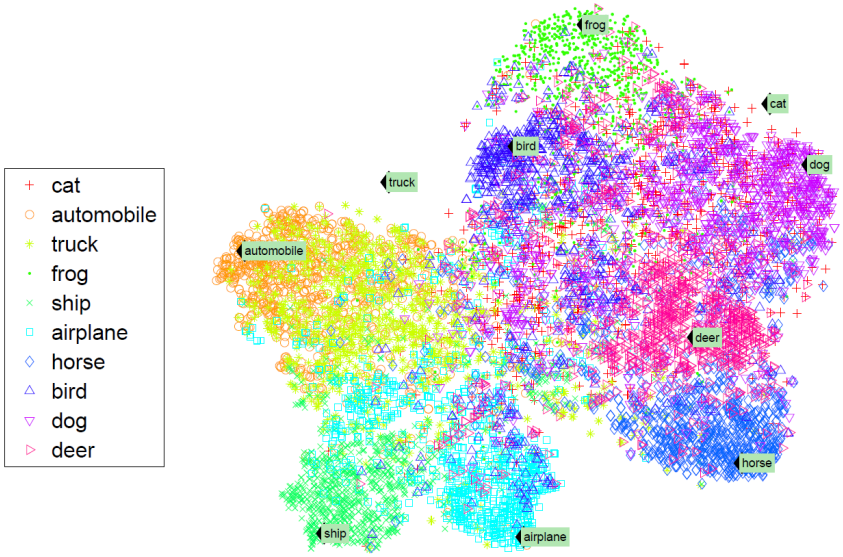
에 syntatic하고 semantic한 정보를 잘 캐치한다.

- $\theta^{(1)}, \theta^{(2)}$: 추정할 모수
- f : 비선형성을 주기 위한 $tanh$

위 식은 역전파나 L-BFGS 방법을 통해 최소화되고, 최소화됨으로써 이 모델은 이미지 feature vector를 50차원의 word semantic space로 매핑시킬 수 있게 된다. 저자들은 이렇게 다른 공간으로 매핑함으로써 시각적인 정보에 나타나는 class의 의미적인 정보를 찾아내고자 했다.

CIFAR-10 dataset

CIFAR-10 이미지로부터 매핑된 50차원의 semantic vector를 tSNE를 이용해 seen class와 unseen class 모두 2차원에 표현한 그림은 아래와 같다.



총 10개의 클래스 중 unseen class는 cat과 truck으로 설정되었다. 8개의 seen class는 꽤나 타이트하게 군집을 이루는 것을 확인할 수 있지만, 2개의 unseen class는 그렇지 않다. 하지만 여기서 신기한 점은 cat은 dog 주변에 있으며, truck은 automobile 주변에 있다. unseen class라도 완전히 동떨어진 곳이 아닌 의미적으로 (semantically) 비슷한 곳에 위치한다는 것이다.

이러한 점은 seen/unseen class를 먼저 구분한 뒤에 이미지를 분류하는 novelty detection의 base idea가 된다.

2. Zero-shot Learning model

이제 이미지 피처가 semantic space로 매핑된 후 벌어지는 과정들을 알아보자.

이미지 분류는 test set X_t 에 속하는 이미지 x가 주어졌을 때 seen class든 unseen class든 특정 class가 등장할 조건부 확률 $p(y|x)$를 예측하는 것이다. 앞절에서 언급했듯이 본 논문에서는 이를 위해 이미지를 semantic vector $f \in F_t$로 매핑했다.

하지만 standard한 분류 모델은 학습에 등장하지 않았던 class를 예측하지 못하므로 저자는 특정 class가 seen인지 unseen인지 구분하는 binary novelty random variable V를 도입한다.

V를 도입함으로써 기존의 주어진 정보로 클래스의 조건부 확률을 예측하는 것은 아래처럼 분리가 가능해진다

$$p(y|x, X_s, F_s, W, \theta) = \sum_{V \in \{s,u\}} P(y|V, x, X_s, F_s, W, \theta)\ P(V|x, X_s, F_s, W, \theta)$$

- X_s : seen class에 해당하는 이미지 set
- F_s : X_s에 대응하는 semantic vector set
- W : seen class들의 semantic word vector set
- θ : 매핑 X_s —> F_s 를 하는 모델의 파라미터

위처럼 V에 대해 분리함으로써 seen과 unseen class의 분류가 가능해졌다. 오른쪽의 식을 통해 두 단계로 분류가 이루어짐을 알 수 있다. 우선 주어진 이미지가 seen인지 unseen인지 예측하고 그 정보를 이용해 class y를 예측한다. 각각의 단계를 자세하게 알아보자.

1. Strategies for Novelty Detection

이미지가 semantic space로 매핑된 후, 이미지가 seen인지 unseen인지 구분하는 novelty detection이 이루어진다. 앞서 tSNE를 이용한 시각화를 보면 unseen class는 seen class 이미지들에 가깝지는 않지만 같은 region에 있는 것처럼 보인다. 따라서 논문에서는 seen, unseen 구분을 위해 outlier detection 방법을 사용한다. 총 두 가지 방법을 제안했으며 모두 semantic space 상에서 계산된다.

- **첫 번째 방법 (isometric Gaussian)**

첫 번째 방법은 정규 분포를 이용한 간단한 방법으로 정규 분포로부터 구해진 이미지가 등장할 확률이 특정 threshold보다 작으면 unseen class로 분류하는 방법이다. 두 번째 방법에 비해 상대적으로 liberal하다는 특징이 있다.

우선 각각의 seen class $y \in Y_s$에 대해,

$$P(x|X_y, w_y, F_y, \theta) = P(f|F_y, w_y) = \mathcal{N}(f|w_y, \Sigma_y)$$

를 구한다.

이 확률은 seen class를 갖는 데이터셋에 대해 하나의 이미지 x가 등장할 확률을 구하는 것이다. 이를 평균은 w_y, 공분산 Σ_y를 갖는 정규 분포 가정을 통해 구한다. 이때 과적합을 방지하기 위해 정규 분포는 isometric으로 제한했다. 만약 확률 값이 threshold T_y보다 작다면, 중심 w_y에서 멀다는 것을 의미하므로 outlier로 간주할 수 있게 된다.

$$P(V = u|f, X_s, W, \theta) := \mathbf{1}\{\forall y \in Y_s : P(f|F_y, w_y) < T_y\}$$

T_y가 작을수록 더 적은 이미지들이 outlier(=unseen)으로 분류된다.

하지만 이 방법의 가장 큰 단점은 outlier에 대한 실제 확률값을 반환하지 않는다는 것이다.

- **두 번째 방법 (Local outlier probaility)**

두 번째 방법은 outlier일 실제 확률을 반환한다. 이 방법은 첫 번째에 비해 이미지를 unseen으로 분류하는 데에 매우 conservative하기에 많은 이미지들을 seen class로 분류하고, 그렇기 때문에 seen class를 예측하는 성능은 좋은 편이다.

밀도 기반으로 outlier score를 메기는 local outlier factor를 활용하는 방법이다. 이 방법은 모든 데이터가 아닌 특정 데이터 주변 k개의 데이터만을 이용하여 outlier를 탐지한다. 논문에서 $k = 20$으로 설정했다. 추가로 표준편차 계수를 나타내는 $\lambda = 3$으로 설정했다.

이미지가 semantic space로 매핑된 형태인 벡터 f와 그 주변 k개의 seen class에 해당하는 이웃 벡터 set을 $C(f)$라고 할 때, *probailistic set distance* pdist 를 구한다.

$$\mathrm{pdist}_\lambda(f, C(f)) = \lambda\sqrt{\frac{\sum_{q\in C(f)} d(f,q)^2}{|C(f)|}}$$

이때 $d(f,q)$는 거리 함수로 논문에서는 유클리디안을 사용했다.

계산된 pdist를 이용해 *local outlier factor* lof 를 구한다. 이름에서도 알 수 있듯이 f 주변의 local한 영역에 대해서만 outlier score를 계산한다. 값이 클수록 outlier로 여길 가능성이 커진다.

$$\mathrm{lof}_\lambda(f) = \frac{\mathrm{pdist}_\lambda(f, C(f))}{\mathbb{E}_{q\sim C(f)}[\mathrm{pdist}_\lambda(f, C(q))]} - 1$$

lof 를 이용해 실제 확률값을 구하기 위해 normalization factor Z를 정의한다. lof 의 기대값이 0이라고 한다면 lof 값의 표준편차로 볼 수도 있다.

$$Z_\lambda(F_s) = \lambda\sqrt{\mathbb{E}_{q\sim F_S}\left[(\mathrm{lof}(q))^2\right]}$$

마지막으로 lof 를 *normalization factor* Z로 나눈 값을 sigmoid 와 비슷한 *Gauss error function* erf 에 넣어줌으로써 특정 semantic vector f에 대한 outlier 확률값 *Local outlier probaility* $LoOP$ 를 구할 수 있게 된다.

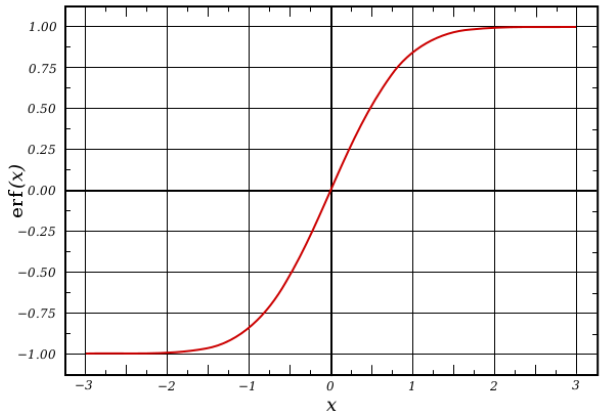
$$LoOP(f) = \max\{0, \mathrm{erf}(\frac{\mathrm{lof}_\lambda(f)}{Z_\lambda(F_s)})\}$$

이 outlier 확률은 이어질 분류에서 seen class와 unseen class의 가중치로 활용될 수 있다.

▼ Gauss error function

확률값을 반환하기 위한 함수로 sigmoid 함수와 비슷하다.

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$$



출처 : Wikipedia

2. Classification

위 novelty dection의 결과에 따라 이미지가 seen 또는 unseen으로 분류되고, 각 경우에 따라 마지막으로 이미지 클래스를 분류하는 방법이 다르다.

- seen class 의 경우, $P(y|V = s, x, X_s)$ 를 구하기 위해 softmax classifier를 사용했다.
- unseen class 의 경우, 각 novel class word vectors에 isometric Gaussian을 가정해 likelihood에 따라 클래스를 분류했다.

4. Experiments

논문에 제시된 모델의 성능을 검증하는 실험에는 10개의 class가 있는 CIFAR-10 데이터셋을 사용했다. 여기서 cat과 truck을 unseen class라고 두고 나머지 8개 class에 대해서만 모델을 학습했다.

CIFAR-10 dataset

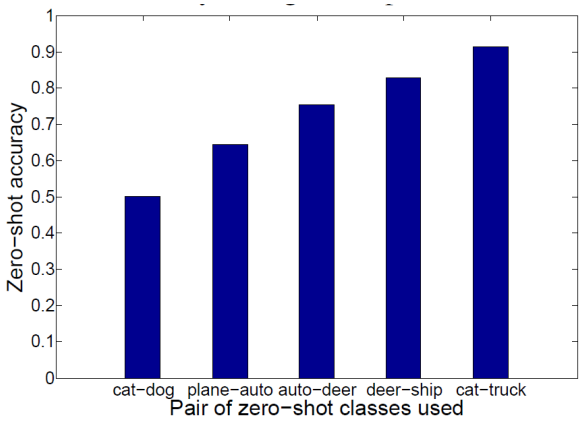
1. Seen and Unseen Class Separately

먼저 일반적인 모델의 분류 성능을 평가하기 위해 seen과 unseen class 각각에 대해 분류 성능을 확인했다.

8개의 seen class는 softmax classifier로 분류했고 82.5%의 정확도를 얻었다.

2개의 unseen class는 isometric Gaussian 기반하여 클래스를 분류했다. 이 isometric Gaussian은 unseen class의 단어 벡터와 semantic space로 매핑된 이미지 사이의 거리를 계산해 가까운 class로 분류한다. unseen을

분류할 때는 unseen과 비슷한 클래스가 seen 클래스에 있을 때 좋은 성능을 보였다. 반면 그렇지 않다면 좋지 않은 성능을 보였다.

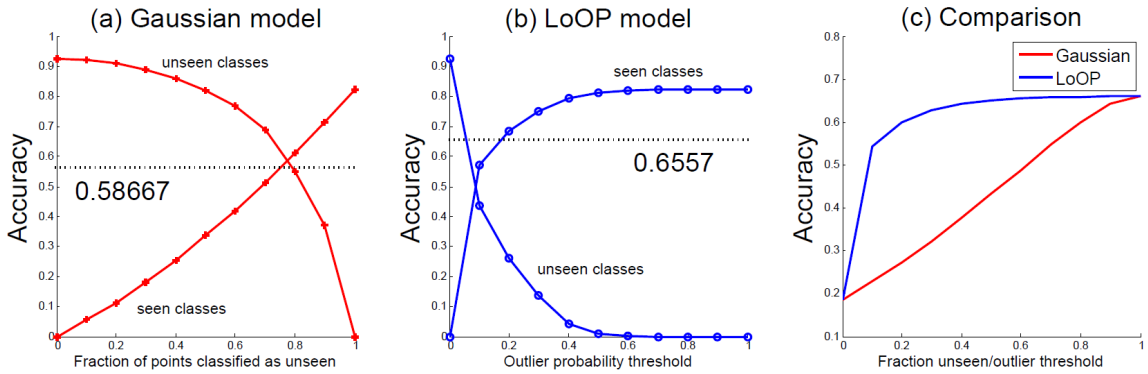


위 사진을 통해서 그 사실을 알 수 있다. 서로 다른 2개 class를 unseen으로 설정했을 때의 분류 정확도를 보여주고 있다. cat-dog 를 unseen으로 설정한 경우, 두 클래스가 서로 유사하며 seen에 이와 비슷한 클래스가 없어 정확도가 낮으며, cat-truck 을 seen으로 설정한 경우, 두 클래스가 서로 구분되며 seen에 이와 비슷한 클래스가 모두 존재하기에 정확도가 낮은 것으로 생각할 수 있다.

2. Influence of Novelty Detectors on Average Accuracy

앞선 경우는 seen과 unseen 각각의 데이터에 대한 분류 결과였다. 이제 두 클래스가 함께 있는 전체 데이터셋에 대한 성능을 보자. 이 과정에서 novelty detection이 사용되기 때문에 novelty detection을 위한 두 방법의 비교가 주된 관심사다.

아래 정확도 그래프를 보자.



novelty dectection을 위한 Gaussian과 LoOP 방법 모두 threshold에 해당하는 x축이 증가할수록 seen class에 대한 분류 성능은 증가하고, unseen class에 대한 성능은 감소하나 그 양상이 다른 것을 볼 수 있다.

Gaussian을 활용한 outlier detection은 unseen으로 분류하는 것에 liberal 하므로, x축이 증가하더라도 unseen class에 대한 분류 성능이 급격하게 나빠지지 않는다. 반면, 앞선 tSNE 시각화 사진에서 seen class와 달리 unseen class는 공간 전체에 넓게 퍼져있으므로 class 간 밀도 차이가 적어 outlier로 분류될 확률이 적다. 따라서 LoOP를 활용한 outlier detection은 unseen에 대한 분류 성능이 급격하게 나빠지는 것을 확인할 수 있다.

따라서 우리가 seen class에 대한 분류 성능을 높일 것인지, unseen class에 대한 분류 성능을 높일 것인지에 따라 novelty detection 방법과 그 threshold를 잘 선택해야 한다.

3. Combining predictions for seen and unseen classes

seen/unseen 의 구분 이후 이미지 클래스를 예측해야 한다. 아래 식을 통해 클래스 예측이 이루어지는 과정을 설명할 수 있다.

$$p(y|x, X_s, F_s, W, \theta) = \sum_{V \in \{s,u\}} P(y|V, x, X_s, F_s, W, \theta)\ P(V|x, X_s, F_s, W, \theta)$$

이는 Bayesian pipeline으로 볼 수 있다. novelty detection에서 $P(V|x, X_s, F_s, W, \theta)$ 을 구함으로써 seen / unseen 에 속할 확률을 구할 수 있고, 이를 $P(y|V, x, X_s, F_s, W, \theta)$의 가중치로서 사용할 수 있다. 마지막으로 V에 대해 marginalizing out하면 이미지 분류를 위한 확률 $p(y|x, X_s, F_s, W, \theta)$을 구할 수 있다.

다만 novelty detection에서 Gaussian 방법을 활용한 detection은 실제 확률을 반환하는 것이 아니므로 cutoff fraction을 조정하여, log 확률값으로 바꿔주는 과정이 필요하다.

4. Comparison to attribute-based classification

본 논문의 novelty detection을 활용한 두 모델 모두 이미지의 Attribute를 side information으로 활용한 attribute-based 분류 모델과 비교해서 더 좋은 성능을 보였다.

Bayesian pipeline (Gaussian)	74.25%
Bayesian pipeline (LoOP)	65.31%
Attribute-based (Lampert et al.)	45.25%

5. Novelty detection in original feature space

이 논문의 핵심 두 가지 중 하나는 이미지 피처를 semantic word space로 매핑시켰다는 것이다. 이미지 피처를 이 space로 매핑시켰을 때, Gaussian을 활용한 novelty detection의 false positive rate는 0.12였다. 하지만 semantic space로의 매핑 없이 원래의 피처 공간에서 novelty detection을 한 경우 false positive rate는 0.78로 매우 안좋은 성능을 보였다.

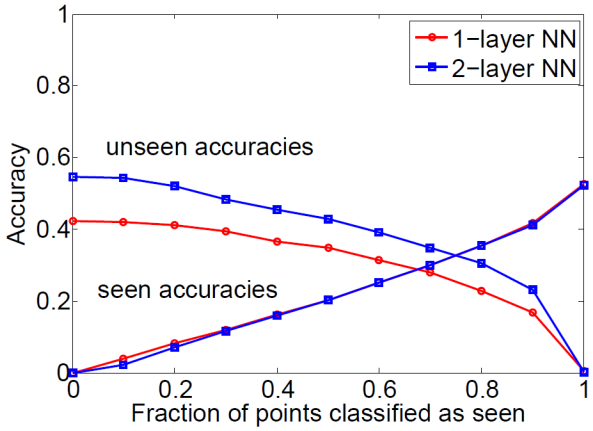
이는 본 논문에서 제시한 semantic space로의 매핑이 seen/unseen 구분에 효과적이었음을 시사하며, 이미지 피처 공간보다 매핑된 공간에서 Gaussian centroid에 의한 이미지들의 클러스터가 잘 형성되었다고 할 수 있다.

6. Extension to CIFAR-100 and Analysis of Deep Semantic Mapping

기존 Experiments에 사용한 CIFAR-10에 100개 클래스의 이미지가 있는 CIFAR-100 데이터를 결합하여 많은 class에 대한 성능도 확인했다.

CIFAR-100 의 100개의 클래스 중에 단어 벡터가 없는 4개를 제외하고 96개를 사용해 총 106개의 클래스가 있는 데이터셋을 만들었다. 이중 6개를 unseen으로 설정하여 분류를 진행했는데, 최고 정확도는 52.7%였다.

클래스 개수가 많이 증가했기 때문에 이미지를 semantic space로 매핑하는 것이 중요해졌고, 이를 위해 매핑 모델에서 1-layer였던 신경망을 2-layer로 증가시켜 성능을 향상시켰다. 아래는 그 그래프이다.



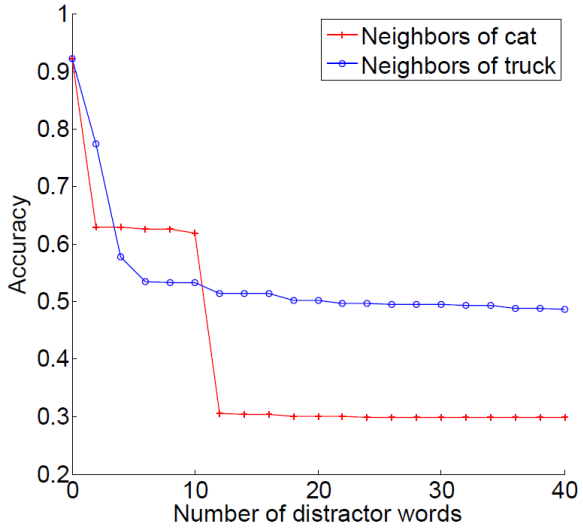
7. Zero-Shot Classes with Distractor Words

마지막 실험은 zero-shot 이미지를 더

1. random distractor noun 추가
2. zero-shot 이미지와 유사한 k개의 단어 추가

1번의 경우 분류 정확도는 크게 차이가 나지 않았지만, 2번의 경우는 많은 차이가 났다. 아래 그래프가 2번의 경우에 대한 정확도 그래프다. 두 zero-shot 이미지 cat과 truck과 유사한 distractor word를 추가해나가면서 정확도를 나타냈는데, 예를 들어 cat에 대한 distractor word는 rabbit, kitten mouse 등이 있다.

예상했던 것처럼 의미적으로 비슷한 단어가 unseen class에 서로 포함된다면 정확도가 급격히 떨어지는 것을 알 수 있다. 하지만 그래프에서 보이듯이 단어가 일정 개수 이상 추가되면 정확도의 큰 변화는 없다.



5. Conclusion

본 논문에서는 ZSL을 위한 모델을 제시했다. 단어가 텍스트 상에서 갖는 의미적인 공간 (semantic space)으로 이미지를 매핑시켜, unseen 이미지에 대해서도 추론이 가능하게끔 한 것이다. 이 모델의 핵심은 두 가지를 꼽을 수 있다.

1. Semantic word vector가 이미지와 텍스트 사이의 knowledge transfer를 담당했다. (cross-modality)
2. 매핑된 semantic space 상에서 seen/unseen을 분류해내는 novelty detection과 Bayesian Framework이 ZSL 성능 향상에 도움이 되었다.